

Normalization



Normalization

- Why do we use Normalization?

Figure 14.4 Example relations for the schemas in Figure 14.3 that result from applying NATURAL JOIN to the relations in Figure 14.2. These may be stored as base relations for performance reasons.

EMP DEPT						
ENAME	SSN	BDATE	ADDRESS	DNUMBER	DNAME	DMGRSSN
Smith,John B.	123456789	1965-01-09	731 Fondren,Houston,TX	5	Research	333445555
Wong, Franklin T.	333445555	1955-12-08	638 Voss,Houston,TX	5	Research	333445555
Zelaya, Alicia J.	999887777	1968-07-19	3321 Castle,Spring,TX	4	Administration	987654321
Wallace, Jennifer S.	987654321	1941-06-20	291 Berry,Bellaire,TX	4	Administration	987654321
Narayan, Ramesh K.	666884444	1962-09-15	975 FireOak,Humble,TX	5	Research	333445555
English, Joyce A.	453453453	1972-07-31	5631 Rice,Houston,TX	5	Research	333445555
Jabbar,Ahmad V.	987987987	1969-03-29	980 Dallas,Houston,TX	4	Administration	987654321
Borg, James E.	888665555	1937-11-10	450 Stone,Houston,TX	1	Headquarters	888665555

EMP PROJ					
SSN	PNUMBER	HOURS	ENAME	PNAME	PLOCATION
123456789	1	32.5	Smith,John B.	ProductX	Bellaire
123456789	2	7.5	Smith,John B.	ProductY	Sugarland
666884444	3	40.0	Narayan,Ramesh K.	ProductZ	Houston
453453453	1	20.0	English,Joyce A.	ProductX	Bellaire
453453453	2	20.0	English,Joyce A.	ProductY	Sugarland
333445555	2	10.0	Wong, Franklin T.	ProductY	Sugarland
333445555	3	10.0	Wong, Franklin T.	ProductZ	Houston
333445555	10	10.0	Wong, Franklin T.	Computerization	Stafford
333445555	20	10.0	Wong, Franklin T.	Reorganization	Houston
999887777	30	30.0	Zelaya,Alicia J.	Newbenefits	Stafford
999887777	10	10.0	Zelaya,Alicia J.	Computerization	Stafford
987987987	10	35.0	Jabbar,Ahmad V.	Computerization	Stafford
987987987	30	5.0	Jabbar,Ahmad V.	Newbenefits	Stafford
987654321	30	20.0	Wallace, Jennifer S.	Newbenefits	Stafford
987654321	20	15.0	Wallace, Jennifer S.	Reorganization	Houston
888665555	20	null	Borg, James E.	Reorganization	Houston



Normalization Avoids

- Duplication of Data
 - The same data is listed in multiple lines of the database
- Insert Anomaly
 - – A record about an entity cannot be inserted into the table without first inserting information about another entity –
Cannot enter a customer without a sales order
- Delete Anomaly
- Update Anomaly
- Frequent Null Values



Functional dependency

- a constraint between two attributes (columns) or two sets of columns
- A set of attributes A *functionally determines* a set of attributes B if the value of A determines a unique value for B
- $A \rightarrow B$ if “for every valid instance of A , that value of A uniquely determines the value of B ”
- or ... $A \rightarrow B$ if “there exists at most one value of B for every value of A ”

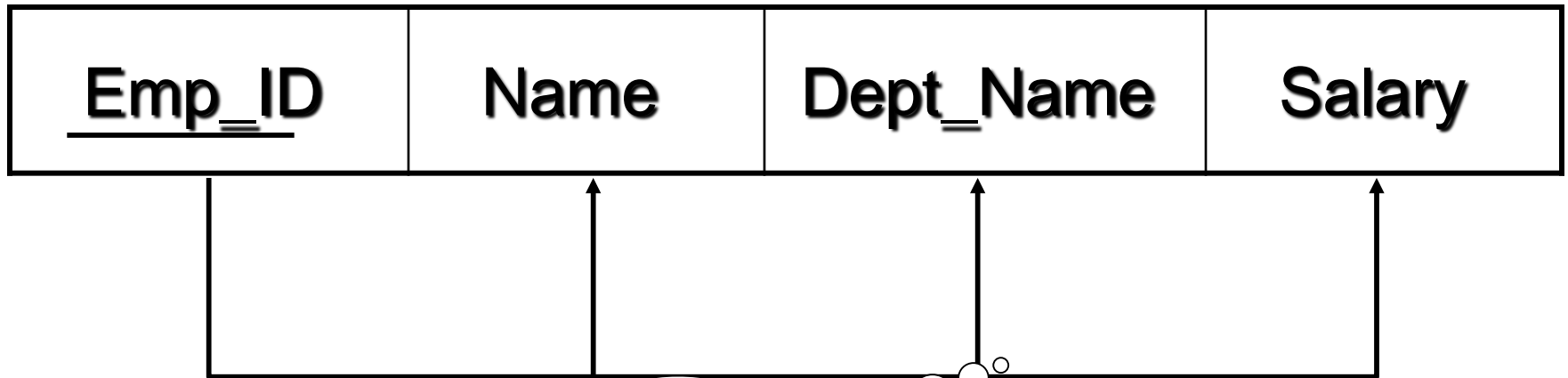
... functional dependency

- some examples
- social security number determines employee name
SSN \rightarrow ENAME
- project number determines project name and location
PNUMBER \rightarrow {PNAME, PLOCATION}
- employee ssn and project number determines the hours per week that the employee works on the project
{SSN, PNUMBER} \rightarrow HOURS
- **So functional dependency is the technical term for *determines***

keys and dependencies

EMPLOYEE1 (Emp_ID, Name, Dept_Name, Salary)

determinant



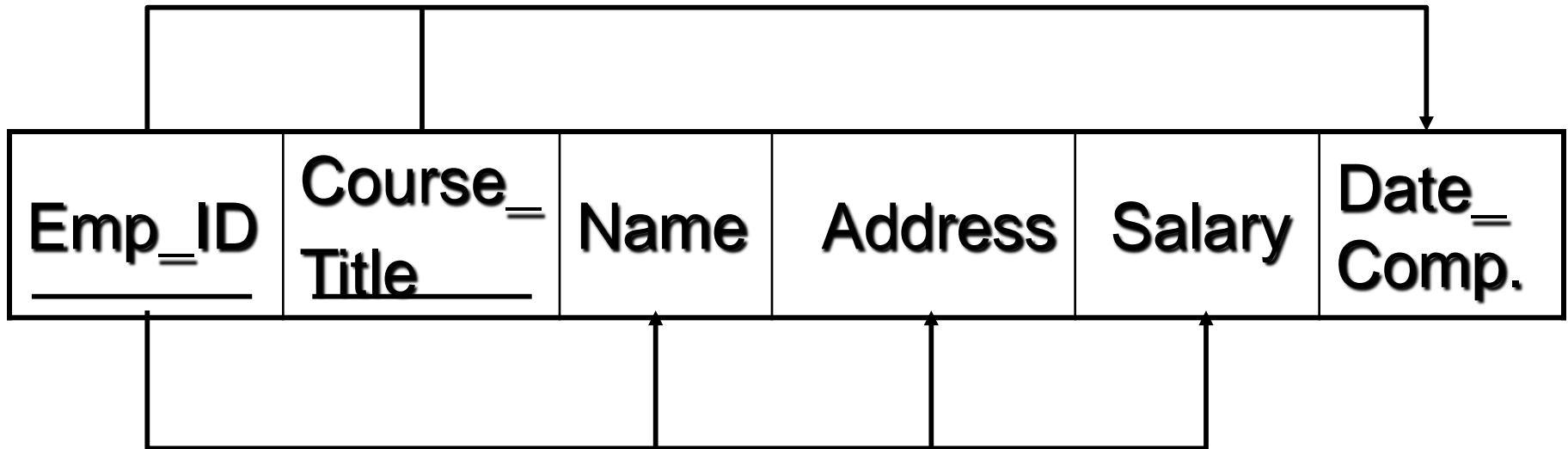
**functional
dependency**

Partial Dependency

- Functional Dependency
 - The value of one attribute in a table is determined entirely by the value of another
- . Partial Dependency
 - A type of functional dependency where an attribute is functionally dependent on only part of the primary key (primary key must be a composite key).

Partial Dependency

EMPLOYEE2 (Emp_ID, Course_Title, Name, Address, Salary, Date_Completed)



not fully functionally
dependant on the primary
key

Attribute types

- In ER Model attributes can be classified into the following types.
- Simple and Composite Attribute
- Single Valued and Multi Valued attribute
- Stored and Derived Attributes
- Complex Attribute



Simple and Composite Attribute

- Simple attribute that consist of a single atomic value.
A composite attribute is an attribute that can be further subdivided. For example the attribute ADDRESS can be subdivided into street, city, state, and zip code. A simple attribute cannot be subdivided. For example the attributes age, sex etc are simple attributes.
- Simple Attribute: Attribute that consist of a single atomic value.
Example: Salary, age etc
- Composite Attribute : Attribute value not atomic.
Example : Address : 'House_no:City:State
Name : 'First Name: Middle Name: Last Name'

Single Valued and Multi Valued attribute

- A single valued attribute can have only a single value. For example a person can have only one 'date of birth', 'age' etc. That is a single valued attributes can have only single value. But it can be simple or composite attribute. That is 'date of birth' is a composite attribute , 'age' is a simple attribute. But both are single valued attributes.
- Multivalued attributes can have multiple values. For instance a person may have multiple phone numbers, multiple degrees etc. Multivalued attributes are shown by a double line connecting to the entity in the ER diagram.
- Single Valued Attribute: Attribute that hold a single value
Example1: Age, City, Customer id
- Multi Valued Attribute: Attribute that hold multiple values.
Example1: A customer can have multiple phone numbers, email id's etc
Example2: A person may have several college degrees

Stored and Derived Attributes

- The value for the derived attribute is derived from the stored attribute. For example 'Date of birth' of a person is a stored attribute. The value for the attribute 'AGE' can be derived by subtracting the 'Date of Birth'(DOB) from the current date. Stored attribute supplies a value to the related attribute.
- **Stored Attribute:** An attribute that supplies a value to the related attribute.
Example: Date of Birth
- **Derived Attribute:** An attribute that's value is derived from a stored attribute.
Example : age, and it's value is derived from the stored attribute Date of Birth.
- **Complex Attribute**
- A complex attribute that is both composite and multi valued.



Definition

- **Normalization:** The process of decomposing unsatisfactory "bad" relations by breaking up their attributes into smaller relations
- **Normal form:** Condition using keys and FDs of a relation to certify whether a relation schema is in a particular normal form

1st Normal Form

- **First Normal Form**
- Separate Repeating Groups into New Tables.
- *Repeating Groups* Fields that may be repeated several times for one document/entity
- Create a new table containing the repeating data
- The primary key of the table (repeating group) is always a **composite key**; Usually document number and a field uniquely describing the repeating line, like an item number.



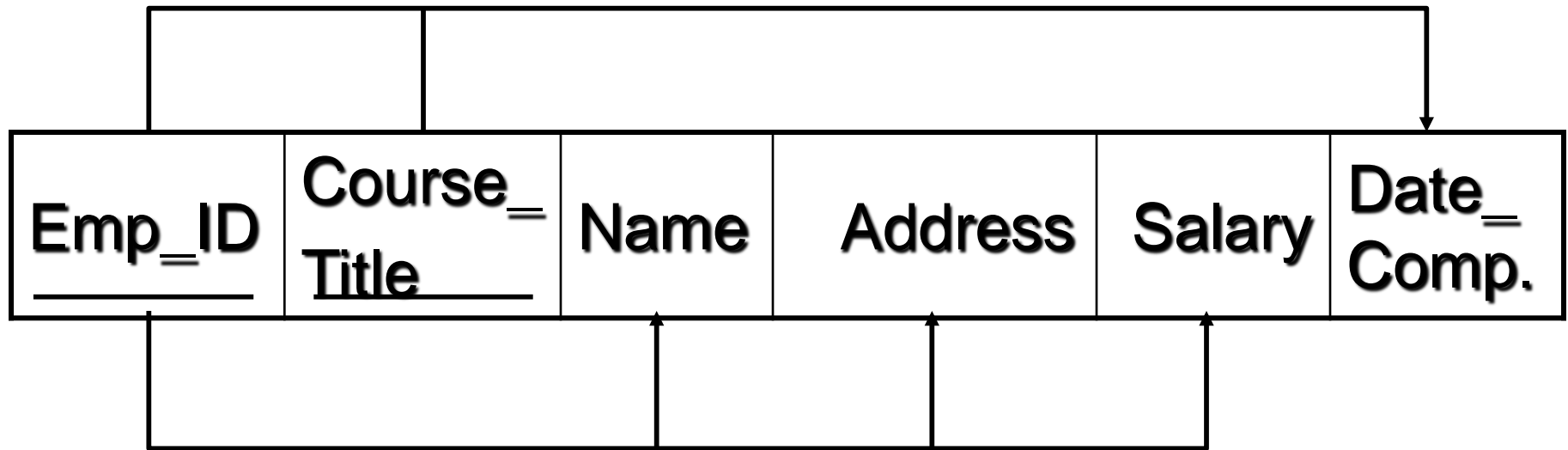
1NF

- relation is in **first normal form** if it contains no multivalued or composite attributes
- remove repeating groups to a new table as already demonstrated, “**carrying**” the PK as a FK
- All columns (fields) must **be atomic**
 - Means : **no repeating items in columns**

2NF

- a relation is in **second normal form** if it is in first normal form AND every nonkey **attribute is fully functionally dependant** on the primary key
- i.e. remove partial functional dependencies, so no nonkey attribute depends on just part of the key

EMPLOYEE2 (Emp_ID, Course_Title, Name, Address, Salary, Date_Completed)



not fully functionally
dependant on the primary
key

2NF

- a relation is in 2NF if it is in 1NF and **any one** of these is true:
 - the PK consists of only 1 attribute
 - all attributes are part of the PK (no nonkey attributes)
 - every non key attribute is functionally dependant on the whole PK

1NF → 2NF

EMPLOYEE2 (Emp_ID, Course Title, Name, Address, Salary, Date_Completed) →

EMPLOYEE1 (Emp_ID, Name, adress, Salary)

and

**EMP_COURSE (Emp_ID, Course Title,
Date_Completed)**

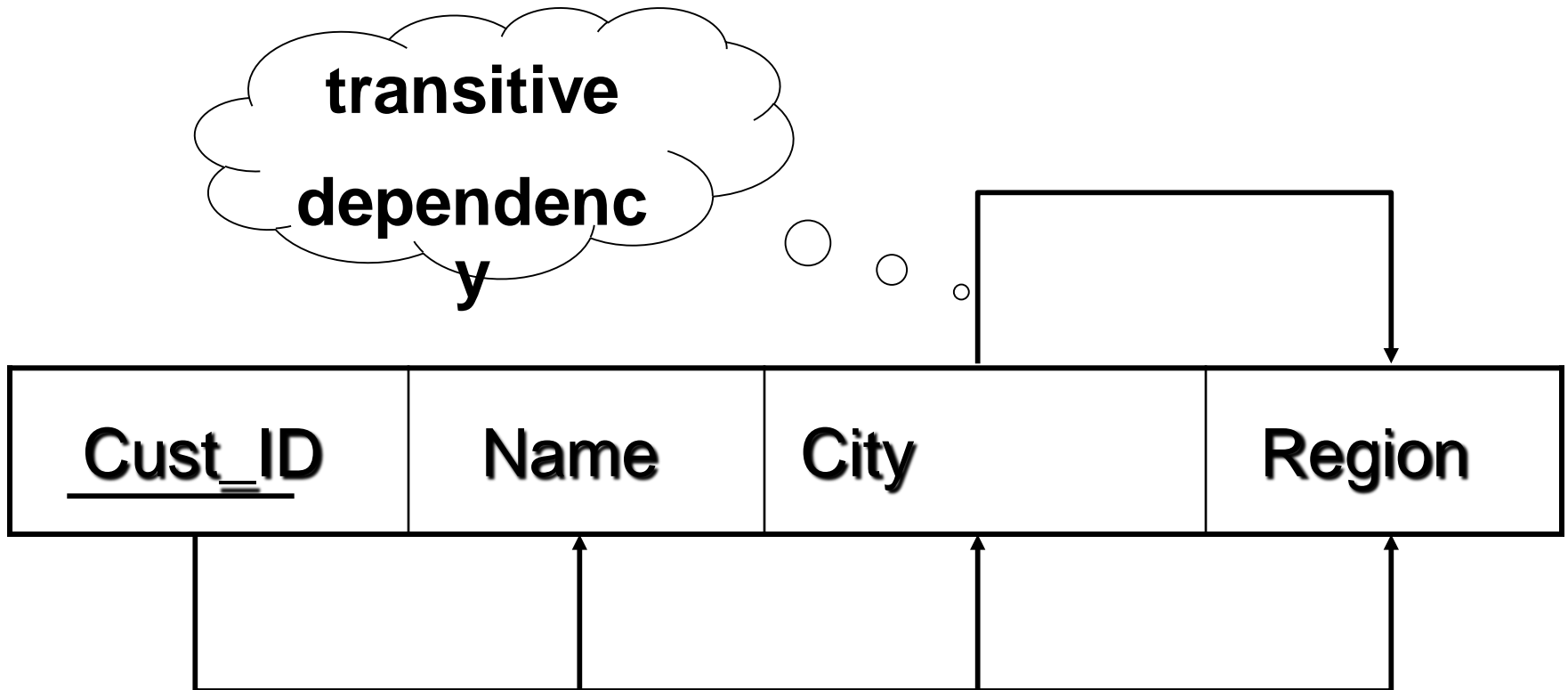
EMPLOYEE1 satisfies condition1

- **EMP_COURSE satisfies condition3**

3NF

- a relation is in **third normal form** if it is in 2NF, AND no *transitive dependencies* exist
- **transitive dependency** is a functional dependency between nonkey attributes

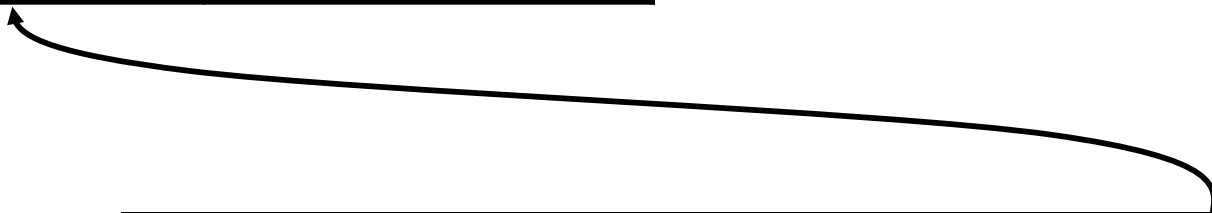
transitive dependency



<u>Cust_ID</u>	Name	city	Region
----------------	------	------	--------

<u>City</u>	Region
-------------	--------

<u>Cust_ID</u>	Name	<u>City</u> -----
----------------	------	-------------------



Normalizing an Example Table

- Unnormalized table:

Student#	Advisor	Adv-Room	Class1	Class2	Class3
1022	Jones	412	101-07	143-01	159-02
4123	Smith	216	201-01	211-02	214-01

First Normal Form: No Repeating Groups

- Another way to look at this problem is with a one-to-many relationship, do not put the one side and the many side in the same table. Instead, create another table in first normal form by eliminating the repeating group (Class#), as shown below:

Student#	Advisor	Adv-Room	Class#
1022	Jones	412	101-07
1022	Jones	412	143-01
1022	Jones	412	159-02
4123	Smith	216	201-01
4123	Smith	216	211-02
4123	Smith	216	214-01

Second Normal Form: Eliminate Redundant Data

- Note the multiple Class# values for each Student# value in the above table. Class# is not functionally dependent on Student# (primary key), so this relationship is not in second normal form.
 - Students

Student# Advisor Adv-Room		
1022	Jones	412
4123	Smith	216

- Registration

Student# Class#	
1022	101-07
1022	143-01
1022	159-02
4123	201-01
4123	211-02
4123	214-01

Third Normal Form: Eliminate Data Not Dependent On Key(**transitive dependency**)

- , Adv-Room (the advisor's office number) is functionally dependent on the Advisor attribute. The solution is to move that attribute from the Students table to the Faculty table, as shown below:

- Students

Student#Advisor	
1022	Jones
4123	Smith

- Faculty

NameRoomDept		
Jones	412	42
Smith	216	42

Sales Order

Fiction Company
202 N. Main
Mahattan, KS 66502

Customer Number: 1001
Customer Name: ABC Company
Customer Address: 100 Points
Manhattan, KS 66502

Sales Order Number: 405
Sales Order Date: 2/1/2000
Clerk Number: 210
Clerk Name: Martin Lawrence

Item Ordered	Description	Quantity	Unit Price	Total
800	widgit small	40	60.00	2,400.00
801	tingimajigger	20	20.00	400.00
805	thingibob	10	100.00	1,000.00
Order Total				3,800.00

Solution

- SalesOrderNo, Date, CustomerNo, CustomerName, CustomerAdd, ClerkNo, ClerkName, ItemNo, Description, Qty, UnitPrice
- **First Normal Form**
- Repetition of Data –Header data repeated for every line in sales order(**ItemNo, Description, Qty, UnitPrice**). Separate them in a new tabel.
- The new table is as follows:
- SalesOrderNo, ItemNo, **Description, Qty, UnitPrice**
- The repeating fields will be removed from the original data table, leaving the following.
- SalesOrderNo, **Date, CustomerNo, CustomerName, CustomerAdd, ClerkNo, ClerkName**



2nd Normal Form

- The new table will contain the following fields:
- ItemNo, Description, UnitPrice
- SalesOrderNo, ItemNo, Qty
- SalesOrderNo, Date, CustomerNo, CustomerName, CustomerAdd, ClerkNo, ClerkName



3rd Normal Form

- CustomerNo, CustomerName, CustomerAdd
- ClerkNo, ClerkName
- SalesOrderNo, Date, CustomerNo, ClerkNo
- Unchanged Tables
- ItemNo, Description, UnitPrice
- SalesOrderNo, ItemNo, Qty

Completed Tables in 3rd Normal Form

- Customers: CustomerNo, CustomerName, CustomerAdd
- Clerks: ClerkNo, ClerkName
- Inventory Items: *ItemNo*, Description, UnitPrice
- Sales Orders: SalesOrderNo, Date, CustomerNo, ClerkNo
- SalesOrderDetail: SalesOrderNo, ItemNo, Qty,
-

Example

Students Sheet

Platform Name : SWE

Platform Description: Software Engineering

Graduate Profile: ALL

Appno	Name	F-code	Faculty	Major	Address	Telno	Found Grade	Total Att. Hrs	Start date
123	Ahmed	SC-phy	Science	Physics	Haram	338684 20	A	600	14 Sep
124	Mona	Eng-cs	Engineering	Computer	Dokki	338974 55,338 97445	B	591	15 Sep
127	Ali	Com-ac	Commerce	Accounting	Nasr City	224159 39	A	550	21 Sep
223	Karim	Med-bio	Medicine	Biochemistry	Sheraton	228684 56	C	600	14 Sep

1st Normal Form

- **Normalization: First Normal Form**
- Separate Repeating Groups into New Tables.
- *Repeating Groups* Fields that may be repeated several times for one document/entity
- Create a new table containing the repeating data
- The primary key of the table (repeating group) is always a composite key; Usually document number and a field uniquely describing the repeating line, like an item number.

1NF :

- **Platform** : pfname , pfdesc , pfgraduate
- **Students**: pfname, appno, name , faculty , major , address, Foundgrade, attd , start_date
- **Std_Tel**: appno, telno

2NF

- **Students:** appno, name , f-code ,faculty , major , address
- **Students_pf:** pfname,appno, Foundgrade, attd , start_date
- **Unchanged Tables**
- **Platform :** pfname , pfdesc , pfgraduate
- **Std_Tel:** appno, telno

3NF

- **Students:** appno, name , major, address
- **Fac_majors:** faculty , major
- **Unchanged Tables**
- **Platform :** pfname , pfdesc , pfgraduate
- **Std_Tel:** appno, telno
- **Students_pf:** pfname,appno, Foundgrade, attd , start_date

Case Study

Relation (s#, country, currency, p#, qty)

where

s# supplier identification number (this is the primary key)

country name of country where supplier is located

currency: Currency of the country of each supplier

p# part number of part supplied

qty quantity of parts supplied to date

In order to uniquely associate quantity supplied (qty) with part (p#) and supplier (s#), a composite primary key composed of s# and p# is used.

Real World - School Data

Student				
First	Parent 1	Parent 2	Application No	
Renee	Ann Jones	Theodore Smith	123	
Lucy	Barbara Mills	Steve Mills	558	
Brendan	Jennifer Jones	Stephen Jones	145	...
City	Postal Code	Home Phone	Program	
Annandale	22003	(703) 323-0893, (703) 3240708	P1	
Annandale	22003	(703) 764-5829	P2	
Fairfax	22032	(703) 978-1083	P3	...
Enrolled	Attended/ days	Birth date	Previous Teacher	Current Teacher
96/97	0	6/25/1983	Hamil	Burke
96/97	0	8/14/1983	Hamil	Burke
96/97	0	6/13/1984	Hamil	Burke